Optimizing Software Inspections

Tom Gilb
Independent Consultant

Software Inspections are now widely known to the software industry, but most organizations do not make the most of them. This is because many people misunderstand and misinterpret Inspection. This article aims to provide some direction as to how to get the most value from your Inspections.

Given adequate management support, Inspection can quickly be turned from the initial chaos phase (20 or more major defects per page) to relative cleanliness (two or fewer major defects per page at exit) within a year. To understand Inspection, two points must be clear:

- Inspection, as I define it (shown with a capital "I"), consists of two main processes: the Defect Detection Process (DDP) and the Defect Prevention Process (DPP) [2].
- A major defect ("major") is a defect that, if it is not dealt with at the requirements or design stage, will probably have an order-of-magnitude or larger cost to find and fix when it reaches the testing or operational stages. On average, the find-and-fix cost for a major defect is one work hour upstream but nine work hours downstream.

DDP finds up to 88 percent of existing major defects in a document on a single pass. This alone is important, but DDP actually achieves greater benefit by teaching software engineers. They go through a rapid, individual learning process, which typically reduces the number of defects they make in their subsequent work by two orders of magnitude.

In addition, DDP can and should be extended to support continuous process improvement. This is achieved by including the associated DPP, which is capable of at least 50 percent (first year, and first project used on) to 70 percent (second or third year) defect frequency reduction, and over 90 percent in the longer term. It has also shown at least a 13-to-1 ratio return on investment (ROI). It is the model for Software Engineering Institute (SEI) Capability Maturity Model (CMM) Level 5.

Raytheon provides a good case study. In six years (end 1988 to end 1994), using DDP combined with DPP, Raytheon reduced rework costs (costs of dealing with preventable errors) from about 45 percent to between 5 percent and 10 percent, and had a 7.7-to-1 ROI. They improved software productivity by a factor of 2.7-to-1, reduced negative deviation from budget and deadlines from 40 percent to near zero, and reduced bug density by about a factor of three [5]. Additional detailed costs and benefits can also be found in [2, 3].

Improving Inspections

Following are some key tips about how to improve your Inspection process and how to begin to achieve the kind of results cited above.

Know Your Purpose for Using Inspection

- *Don't misuse Inspection as a "clean-up" process. Use it to motivate, teach, measure, control quality, and to improve your processes.* Most people seem to think Inspection is for cleaning up bad work, bugs, and defects. However, the greatest payback comes when Inspection improves future work, i.e., reduces defect injection. Ensure that your Inspection process fully supports the aspects of *teaching* and *continuous process improvement.*

For continuous process improvement, integrate DPP into conventional Inspections (see [2] for specific details). DPP needs to be practiced early. CMM Level 5 is too important to be put off until later—you need to do it from the beginning.

- *Plan Inspections to address your Inspection purposes.* I have listed over 20 distinct purposes for using Inspection, e.g., document quality, removing defects, job training, motivation, helping a document author, improving productivity, and reducing maintenance costs [3, 5]. Each Inspection will address several of these purposes to varying degrees. Always be aware which purposes are valid for your current Inspection, and formally plan to address them, e.g., by choosing checkers with relevant skills and giving them appropriate checking roles.

Measure Inspection Benefits

- *Measure your benefit from using Inspections.* Inspection should always be highly profitable, e.g., 10-to-1 ROI. If not, it is time to adjust the Inspection process or stop it. Benefits to be measured include rework costs, predictability, productivity, document quality, and ROI [1].

Make Intelligent Decisions on What You Choose to Inspect

- *Use Inspection on* any *technical documentation.* Most people think Inspection is about source code inspection. It *started* there in the early 1970s; however, once you realize that Inspection is not (in the medium term) a clean-up process, you can use it to measure and validate any technical documentation—even technical diagrams.
- *Inspect upstream first.* By the end of the 1970s, IBM and Inspection-method founder Michael Fagan recognized that defects and the profitable use of Inspection lay upstream in the requirements and design areas. Bellcore found that 44 percent of all bugs were due to defects in requirements and design reaching the programmers [4]. If your systems start with contracts, management and marketing plans, you must start your inspection activity there, where the problems start.

One of the most misunderstood dictums to come from the early inspections was "no managers." Wrong! That was in the days when Inspection dealt

16 *CROSSTALK* The Journal of Defense Software Engineering

March 1998

only with source code. Management inspections may be the most useful you will ever do. I have extremely positive experiences using Inspection with top managers on contracts, marketing, and product development plans.

• *Check the significant portions of the material—avoid checking commentary.* Most organizations waste time checking nonsignificant document areas that do not translate into a final product; defects in such areas cannot trigger major consequences. The result of this indiscriminate checking of trivia, at an optimum rate, is 90 percent minor defects and a 90 percent waste of time. It is like checking comments for 90 percent of the time instead of real code.

We have found that it pays to have a general technical documentation rule that technical authors must distinguish between text (or diagrams) that can translate to serious downstream costs (noncommentary or "meat") and less-important areas (commentary or "fat"). This distinction can be done, for example, by using italics for the fat. Some of our clients have even created Microsoft Word macros to count the volume of noncommentary text and print it on the first page. Of course, the checker is allowed to scan and reference the commentary words but is not obligated to check them against all sources, rules, and checklists. It is not worth it.

• *Use sampling to understand the quality level of a document.* It is neither necessary nor desirable to check all pages of long documents. Representative samples will probably tell you whether a document is clean enough to exit at, for example, 0.2 majors per page maximum remaining.

The main purpose of Inspection is *economic*—to reduce lead time and people costs caused by downstream defects—not primarily to clean bugs. As in Harlan Mills' IBM "Cleanroom" method, bugs should be cleaned up or avoided using disciplines such as Watts Humphrey's Personal Software Process (PSP), Structured Programming, and Continuous Improvement and Verification. If all this works as it should, cleaning is unnecessary, and sampling tells you if it is *economically safe* to release the document. Perfection is not required—

it costs infinite resources and is dangerous as a guiding concept.

• *Inspect early and often while a document is being written.* Inspection after a large (100 pages or more) technical document has been "finished" is a common, bad idea. If the process that generates the document is faulty, discover it early and put it right. This saves time and corrects bad processes before they damage your schedule too much.

Focus on Finding the Majors

• *Check at your organization's optimum rates to find major defects.* This is the big one! Most everybody, including so-called teachers of Inspection, manage to miss this point. Or worse, their suggested checking rates are 10 times optimum speed. Optimum *checking* rate is not optimum *reading* rate. Checking in real Inspections involves checking a page against all related documents. This can involve up to 10 or 20 source documents of large size, checklists, and standards. You have to check a *single* line against *many* sources, and it takes time.

Adequate Inspection statistics can prove your employees have a clear, dramatic, and consistent optimum checking rate on specific document types. This ranges between 0.2 and 1.8 pages of 300 noncommentary words per checking hour. At Raytheon, it was about 20 plus or minus 10 lines per hour (0.3 pages). Unfortunately, in spite of their own data, Raytheon suggested rates of about 100 to 250 lines per hour. This was probably because they had finite deadlines and did not understand sampling.

As the checking speed moves toward an optimum speed for effectiveness of finding major defects, the curve for optimum checking rate moves dramatically upward in terms of major defects identified per logical page. The optimum may seem slow, but considering the amount of checking you have to do, it is fast. The main point is that there is a best speed at which to check, and you will be operating at low checking productivity if you fail to heed it.

Note that the optimum checking rate applies both to the checking carried out *before* and to the optional checking carried out *during* the logging meeting.

This second check will produce roughly an additional 15 percent defects. You do not need this extra checking if the document is found clean enough to exit as a result of initial checking sampling or if it is so polluted that you have to do a major rewrite anyway.

• *Define a major defect as "possible larger costs downstream."* It does not matter if a defect is not precisely a bug or if it is visible to a customer. If it *can* potentially lead to significant costs if it escapes downstream, classify it as a major and treat it with due respect.

You can help people identify majors by using checklists that specify how to find them. (Note: checklists are only allowed to help interpret the "rules," which are the official standards for writing a given document, and which define defects). Use symbols: "M" for major and "m" for minor after the checklist questions or rule statement. I also often find it useful to use "S" for super major or showstopper (a defect where the downstream effect could be an order of magnitude bigger than an average major). Super majors can be highlighted for management attention.

• *Log only major defects.* This helps avoid the "90 percent minor" syndrome that often hampers Inspection. Employees waste time identifying 90 percent minor defects unless strongly redirected. There are 18 tactics that shift your focus from minor to major defects ([2], pp. 75-76). For example, allow only ideas for finding majors onto the rules and checklists, log only majors at a meeting, and calculate ROI for Inspections only based on majors. A clear message must be given to not waste time on minor defects.

Apply Good Practice When Leading Inspections

• *Use serious entry conditions, e.g., numeric quality of sources.* We are in such a big hurry to waste our own time. Many do not have the discipline to set up and respect entry conditions that prevent wasting time—but you must.

One of the most important entry conditions is to mandate the use of upstream source documents to inspect a "product" document. It is a mistake to try to use the experts' memory abilities

instead of updated, Inspection-exited source documents. It is a farce to use source documents with the usual uncontrolled, uninspected, unexited, 20 or more major defects per page to check a product document. It is silly to allow a product document author to use a bad quality source document to generate a product. Inspection does not have to repeat that silliness.

You can ascertain the state of a source document through inexpensive sampling. A half day on a few pages is a small price to pay to know the state of a document that could destroy the quality of all your work and your project.

Another serious entry condition is to do a cursory check on the product document and return it to the author when it is anything less than a quality piece of work bursting at the seams to exit (based on a cursory check that reveals few remaining defects). For example, if while planning the Inspection, the team leader performs a 15-minute cursory check that shows up a few major defects on a single page, it is time for a word with the author in private. Pretend the document was never seriously submitted. Certainly do not waste the time of your team to confirm shoddy work.

In short, learn which entry conditions you need to set, then take them seriously. I would like to see management lead by understanding the importance of this instead of ignorantly thwarting engineers' attempts to do reasonable work by insisting Inspections proceed when the entry conditions have not been met.

• *Make sure you have excellent standards to identify defective practices.* Inspection requires that good work standards be in place. Standards provide the rules for the author when writing technical documents and then for the Inspection process to subsequently check against. Standards must be built by hard experience; they need to be brief and to the point, be monitored for usefulness, and must be respected by the troops. They must not be built by outside consultants or dictated by management. They must be seen as the tool to enforce the necessary lessons of professional practice on the unwary or unwilling.

• *Check against source and kin documents; check them for defects, too.* Because of potentially poor quality control practices and craftsmanship, and because Inspection is imperfect on first pass (30 percent to 80 percent effective), focus on major defects that persist in *source* documents used to produce the document under Inspection, and in kin documents derived from the same source documents. For example, if a functional specification was the product document requiring Inspection, there should be a requirements document as one of the source documents and a testing document as one of the kin documents.

Most people overfocus on the product document, i.e., the document you are inspecting and evaluating for exit. You should probably be finding 25 percent of your total defects *external* to the product document.

• *Use the optimum number of people on a team to serve the current purpose of Inspection, e.g., effectiveness, efficiency, and training.* For 13 years, one large U.S. telecommunications company had 12 to 15 people on each inspection team because each "had" to be there to protect territorial interests. There seemed to be no motivation to cut these costs.

The number of people who are needed on a specific Inspection team is a function of your purposes. Monitor the results of varying team sizes to discover your optimum. If you measure your own Inspection experiences, you will find that effectiveness at finding major defects uses four to six people, efficiency (effect over cost) needs two to four people, and only teaching as a purpose justifies larger numbers.

• *Allocate special defect searching roles to people on the team.* Each person on an Inspection team should be finding *different* defects. Much like a coach on a ball team, the Inspection team leader should assign specialist roles to team members, e.g., identify time and money risks, check against corporate standards for engineering documentation, and check security loopholes.

• *Use checking data (such as pages checked, majors found, time used, and checking rate) from individual checkers to decide whether it is worth holding a logging meet-*

ing. Older types of inspection plunge into the logging meeting without forethought, and consequently waste a lot of time. We have developed a process of logging meeting entry evaluation before going ahead with the logging meeting. We collect data from checkers about checking rates and major-issue density. (To avoid personal conflict, issues—not defects—are logged during the logging meeting. An issue may or may not become a defect.) Based on this data, we make a series of decisions about the logging meeting. Most critical is whether a meeting is necessary. Other decisions include whether to log minors, whether to continue checking, and what is the likely optimum checking rate.

• *Use the individual checkers' personal notes instead of proper meeting defect logs when the major issue volume is (nonexit level) high, or when there is a large number of minor defects.* Checkers should not be required to use any particular method to make notes during checking. But most of them mark a paper document (some use an electronic document) with an underline or circle, or they highlight offending words. It is important that they also note, against offending words, *exactly* which rule was broken (the issue). To note major or minor is less important if all issues found are majors.

Whenever there is a higher volume of issues than would indicate allowable exit, you can happily, with author agreement, return these "scratchings" to the author rather than pedantically log them like good bureaucrats. Authors need to rewrite and resubmit in these cases, using this information to correct their work processes (usually to take sources and rules more seriously).

• *At logging meetings, avoid discussions and avoid suggesting fixes.* Inspection is not for talkers and quibblers—it is for professionals committed to making maximum, meaningful progress on the project. You can have a good time, but not by idle gossip and insults. You are there to measure, not to wear each other out, or get drowned in unprofitable bureaucratic games.

• *Use serious exit conditions, e.g., "maximum probable remaining major defects per page is 0.2 for exit."* Exit conditions, if

correctly formulated and taken seriously, can be crucial. It is ridiculous and sad to have the customary "vote" to accept a document when the logged defects are fixed—this completely ignores the known factor of *remaining unfound defects,* which is computable and verifiable from past data and experience.

Remember that Inspection processes, as other testing processes, have a maximum effectiveness for a single pass in the range of 30 percent to 88 percent of existing defects. If the maximum probable remaining defect density is a high-quality low count, e.g., 0.2 majors per page, it matters little if the detected defects are removed; the document is clean enough (economically speaking) to exit without fixing those identified.

If defect density is high, e.g., 20 or more majors per page (quite common), undetected defects, at say 50 percent effectiveness, are more than enough to make exit uneconomical. Ten majors remaining per page in a 100-page document would result in 9 x 10 x 100 hours of additional project work to clean them up by testing and discovery in the field. It costs an order of magnitude less to find them now. Admittedly, it is only the lesser of two evils—you should wish to *prevent* them using DPP rather than have to *clean them up* early using DDP.

Management must understand the large-scale economics of this, making clear policy about the levels of major defects per page that will be allowed to escape ([2] pp. 430-431). The consequences of poor policy here should be deducted from management's pay!

Ensure You Have Provided Adequate Training and Follow-up

• *Give Inspection team leaders proper training, coaching after initial training, formal certification, statistical follow-up, and if necessary, remove their "license to inspect."* Proper training of team leaders takes about a week (half lectures and half practice). Formal Inspection team leader certification (an entry condition to an Inspection) should be similar to that for pilots, drivers, and doctors—based on demonstrated competence after training.

Leaders who will not professionally carry out the job, even if it is because their supervisor is pressuring them to cut corners, need to have their license revoked. If you take professional leadership seriously, your players will take Inspection seriously. Ensure there is an adequate number of trained people to support your Inspections—at least 20 percent of all professionals. Some clients train all of their engineers.

Give Visibility to Your Inspection Statistics and Support Documentation

• *Put your Inspection artifacts on a company Web site.* If you have an Intranet, all relevant Inspection artifacts, standards, experiences, statistics, and problems should be placed on a corporate-wide site as soon as possible.

• *Build or buy an automated software tool to process Inspection basic data.* Use automated software tools to capture summary data and to present trends and reports [7]. Inspection quickly generates a lot of data that is fundamental and useful to managing the process. It is vital that good computer support be given early so the process owners and management take the data seriously and so that early champions are not overwhelmed.

The key distinction between Inspections and other review processes is the use of data to manage them. For example, optimum checking rates must be established early and updated as they change through continuous improvement. It also is vital to statistically see the consequences of inadequate exit levels (too many major defects floating downstream), which then are caught with expensive testing processes. Locally made spreadsheet software is a good start-up tool.

• *Plan Inspections well using a master plan.* We have developed a one-page master plan ([2] p. 401) that goes far beyond the conventional "invitation." We document the many supporting documents needed, assign checkers their special defect-searching roles, and carefully manage rates of checking and the total checking time needed. We establish the formal purpose(s) of this specific Inspection, which vary. We establish a team numeric stretch goal for this Inspection and a strategy to help attain it.

A good master plan avoids senseless bureaucracy and lays the groundwork for intelligent Inspections.

Continuously Improve Your Inspection Process

• *Use Defect Prevention Process on the Inspection process.* Finally, recognize that systematic continuous improvement of the Inspection process is necessary. Initially, this is required not only to improve the process but also to learn the Inspection process properly and to tailor it to your organization. ◆

About the Author

Tom Gilb is a freelance consultant. Born in California, he has been a resident of Norway since joining IBM, there, in 1958. He has the Department of Defense as his favorite worthy cause.

E-mail: Gilb@ACM.org

References
1. Raytheon Defense Electronics, http://www.sei.cmu.edu/products/publications/95.reports/95.tr.017.html.
2. Gilb, T. and D. Graham, *Software Inspection*, Addison-Wesley Longman, London, England, 1993.
3. http://www.stsc.hill.af.mil/SWTesting/gilb.html.
4. Pence, J. L. Pete and Samuel E. Hon III, Bellcore, Piscataway, N.J., "Building Software Quality into Telecommunications Network Systems," *Quality Progress*, October 1993, pp. 95-97.
5. The Raytheon Report, http://www.sei.cmu.edu/products/publications/95.reports/95.tr.017.html. Also see http://www.Result-Planning.com and http://www.stsc.hill.af.mil/SWTesting/gilb.html.
6. A set of slides that correspond to this article should be at the Washington, D.C. Spin site under "Old Lectures," http://www.software.org/DCSpin.
7. Software Development Technologies, Software Inspections Automation, Edward Kit, sdt@sdtcorp.com, http://www.sdtcorp.com.